# Database Applications Exam (Jun 2015) – Photography

Your exam consists of several parts, explained below. You may work independently on each exam part. Submit your solutions as a single **ZIP file holding the full source code**, without any libraries and compiled binaries.

## Part I – Query Existing Database, Import and Export Data

You are given a **MS SQL Server database "Photography"** holding users, photographs, albums, categories, cameras, lenses, manufacturers, available as **SQL script**. Your task is to write a few data-driven applications in C# for importing data, querying data and exporting data from the database.

### Entity Framework Mappings (Database First)

Create an **Entity Framework (EF) data model** of the existing database (map the database tables to C# classes). Use the "**database first**" model in EF. To test your EF data model, **list all camera’s manufacturer + model**. Order them by “**manufacturer + model**”.

Example:

|  |
| --- |
| Canon 5DS  Canon EOS 1200D  Canon EOS 750D  … |

3 score

Make sure all navigation properties have **good (self-describing) names**.

2 score

### Export the Manufacturers and Cameras as JSON

Write a **C# application** based on your EF data model for **exporting all manufacturers along with their cameras** in the following JSON format:

|  |
| --- |
| **manufactureres-and-cameras.json** |
| [  { "manufacturer": "Agfa", "cameras": [] },  { "manufacturer": "Canon", "cameras": [  { "model": "5DS", "price": 3899.0000 },  { "model": "EOS 1200D", "price": 849.0000 },  { "model": "EOS 750D", "price": 749.0000 },  { "model": "EOS 760D", "price": 1298.0000 },  { "model": "EOS 7D Mark II", "price": 2248.0000 },  { "model": "EOS M3", "price":null }  …  ]  }  …  ] |

Write the output in a JSON file named **manufactureres-and-cameras.json**. Include in the output the manufacturers with no cameras (if any). The code indentation in the JSON file is not important.

6 score

Order the **manufacturers** and the **cameras in each manufacturer alphabetically by name**.

2 score

For better performance, ensure your program executes a **single DB query** and retrieves from the database only the required data (without unneeded rows and columns).

2 score

### Export Photographs as XML

Write a **C# application** based on your EF data model for **exporting all photographs, category and equipment** in a XML file named photographs.xml in the following XML format:

|  |
| --- |
| **photographs.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <photographs>  <photograph title=".">  <category>Portrait</category>  <link>http://photo-forum.net/i/1920515</link>  <equipment>  <camera megapixels="24">Canon EOS 760D</camera>  <lens>Canon EF 15mm f/2.8 Fisheye</lens>  </equipment>  </photograph>  <photograph title="Angel eyes">  <category>Portrait</category>  <link>http://photo-forum.net/i/1919347</link>  <equipment>  <camera megapixels="24">Canon EOS 760D</camera>  <lens>Canon EF 15mm f/2.8 Fisheye</lens>  </equipment>  </photograph>  <photograph title="Dog">  <category>Street</category>  <link>http://photo-forum.net/i/1920281</link>  <equipment>  <camera megapixels="24">Canon EOS 750D</camera>  <lens price="799.00">Canon EF 17-40mm f/4.0L USM</lens>  </equipment>  </photograph>  …  </photographs> |

**Each photograph** should have **category**, **link** and **equipment** (camera and lens). Each **camera** should have **megapixels** (as attribute). Each **lens** should have **price** (as attribute) if there is a price. Use an XML parser by choice.

10 score

Order the **photographs** by **title** (in ascending order).

3 score

For better performance, ensure your program executes a **single DB query** and retrieves from the database only the required data (without unneeded rows and columns).

2 score

### Import Manufacturers and Lenses from XML

Write a **C# application** based on your EF data model for **importing manufacturers and lenses**. The application should process **multiple requests** and write **logs** for each operation at the console.

The input comes from an XML file manufacturers-and-lenses.xml in the following format:

|  |
| --- |
| manufacturers-and-lenses**.xml** |
| <?xml version="1.0" encoding="utf-8" ?>  <manufacturers-and-lenses>  <manufacturer>  <manufacturer-name>Fujifulm</manufacturer-name>  <lenses>  <lens model="XC 16-50mm F3.5-5.6 OIS" type="Zoom lens" price="198" />  <lens model="XC 50-230mm F4.5-6.7 OIS" type="Telephoto zoom lens" price="299" />  <lens model="XF 10-24mm F4 R OIS" type="Wideangle zoom lens" price="799" />  <lens model="XF 14mm F2.8 R" type="Wideangle prime lens" price="899" />  <lens model="XC 16-50mm F3.5-5.6 OIS II" type="Zoom lens" />  <lens model="XC 50-230mm F4.5-6.7 OIS II" type="Telephoto zoom lens" />  <lens model="XF 16-55mm F2.8 R LM WR" type="Zoom lens" price="1199" />  <lens model="XF 16mm F1.4 R WR" type="Wideangle macro prime lens" price="999" />  <lens model="XF 18-55mm F2.8-4 R LM OIS" type="Zoom lens" price="699" />  </lenses>  </manufacturer>  <manufacturer>  <manufacturer-name>Panasonic</manufacturer-name>  <lenses>  <lens model="Lumix G Vario 7-14mm F4 ASPH" type="Wideangle zoom lens" price="825" />  <lens model="Lumix G Fisheye 8mm F3.5" type="Wideangle fisheye prime lens" price="641" />  <lens model="Lumix G Macro 30mm F2.8 ASPH Mega OIS" type="Macro prime lens" />  <lens model="Lumix G Vario 14-45mm F3.5-5.6 ASPH OIS" type="Zoom lens" price="392" />  <lens model="Lumix G Vario 45-200mm F4-5.6 OIS" type="Telephoto zoom lens" price="240" />  <lens model="Lumix G X Vario 35-100mm F2.8 OIS" type="Telephoto zoom lens" price="950" />  </lenses>  </manufacturer>  <manufacturer>  <manufacturer-name>Olympus</manufacturer-name>  <lenses>  <lens model="M.Zuiko ED 75-300mm 1:4.8-6.7 II" type="Telephoto zoom lens" price="499" />  <lens model="M.Zuiko Digital ED 40-150mm F2.8 Pro" type="Telephoto zoom lens" price="1399" />  <lens model="M.Zuiko Digital ED 12-40mm 1:2.8 PRO" type="Zoom lens" price="864" />  <lens model="M.Zuiko Digital ED 8mm F1.8 Fisheye Pro" type="Wideangle fisheye prime lens" price="946" />  <lens model="Zuiko Digital ED 50-200mm 1:2.8-3.5 SWD" type="Telephoto zoom lens" price="1180" />  <lens model="Zuiko Digital ED 70-300mm 1:4.0-5.6" type="Telephoto zoom lens" price="518" />  <lens model="Zuiko Digital ED 150mm 1:2.0" type="Telephoto prime lens" />  </lenses>  </manufacturer>  </manufacturers-and-lenses> |

The input XML holds a sequence of requests given in the **<manufacturer>…</manufacturer>** elements.

The element "**manufacturer-name**" is required. The specified manufacturer should be created in the database, if it does not exist.

The elements "**lenses**" and "**lens**" are optional. The specified lenses should be created in the database, if they do not exist. Note that lens "**model**" and “**type**” are mandatory, but lens "**price**" is optional. A lens is considered existing in the database when it is matched by model.

The **output** should be printed on the console in the following format:

|  |
| --- |
| Processing manufacturer #1 ...  Created manufacturer: Fujifulm  Created lens: XC 16-50mm F3.5-5.6 OIS  Created lens: XC 50-230mm F4.5-6.7 OIS  Created lens: XF 10-24mm F4 R OIS  Created lens: XF 14mm F2.8 R  Created lens: XC 16-50mm F3.5-5.6 OIS II  Created lens: XC 50-230mm F4.5-6.7 OIS II  Created lens: XF 16-55mm F2.8 R LM WR  Created lens: XF 16mm F1.4 R WR  Created lens: XF 18-55mm F2.8-4 R LM OIS  Processing manufacturer #2 ...  Existing manufacturer: Panasonic  Created lens: Lumix G Vario 7-14mm F4 ASPH  Existing lens: Lumix G Fisheye 8mm F3.5  Created lens: Lumix G Macro 30mm F2.8 ASPH Mega OIS  Existing lens: Lumix G Vario 14-45mm F3.5-5.6 ASPH OIS  Created lens: Lumix G Vario 45-200mm F4-5.6 OIS  Created lens: Lumix G X Vario 35-100mm F2.8 OIS  Processing manufacturer #3 ...  Existing manufacturer: Olympus  Created lens: M.Zuiko ED 75-300mm 1:4.8-6.7 II  Created lens: M.Zuiko Digital ED 40-150mm F2.8 Pro  Created lens: M.Zuiko Digital ED 12-40mm 1:2.8 PRO  Existing lens: M.Zuiko Digital ED 8mm F1.8 Fisheye Pro  Existing lens: Zuiko Digital ED 50-200mm 1:2.8-3.5 SWD  Created lens: Zuiko Digital ED 70-300mm 1:4.0-5.6  Existing lens: Zuiko Digital ED 150mm 1:2.0 |

The above result should be produced when the program is executed over a **clean database** (as it is initially given in the database create script).

If the program is executed twice, its **second output** should be as follows:

|  |
| --- |
| Processing manufacturer #1 ...  Existing manufacturer: Fujifulm  Existing lens: XC 16-50mm F3.5-5.6 OIS  Existing lens: XC 50-230mm F4.5-6.7 OIS  Existing lens: XF 10-24mm F4 R OIS  Existing lens: XF 14mm F2.8 R  Existing lens: XC 16-50mm F3.5-5.6 OIS II  Existing lens: XC 50-230mm F4.5-6.7 OIS II  Existing lens: XF 16-55mm F2.8 R LM WR  Existing lens: XF 16mm F1.4 R WR  Existing lens: XF 18-55mm F2.8-4 R LM OIS  Processing manufacturer #2 ...  Existing manufacturer: Panasonic  Existing lens: Lumix G Vario 7-14mm F4 ASPH  Existing lens: Lumix G Fisheye 8mm F3.5  Existing lens: Lumix G Macro 30mm F2.8 ASPH Mega OIS  Existing lens: Lumix G Vario 14-45mm F3.5-5.6 ASPH OIS  Existing lens: Lumix G Vario 45-200mm F4-5.6 OIS  Existing lens: Lumix G X Vario 35-100mm F2.8 OIS  Processing manufacturer #3 ...  Existing manufacturer: Olympus  Existing lens: M.Zuiko ED 75-300mm 1:4.8-6.7 II  Existing lens: M.Zuiko Digital ED 40-150mm F2.8 Pro  Existing lens: M.Zuiko Digital ED 12-40mm 1:2.8 PRO  Existing lens: M.Zuiko Digital ED 8mm F1.8 Fisheye Pro  Existing lens: Zuiko Digital ED 50-200mm 1:2.8-3.5 SWD  Existing lens: Zuiko Digital ED 70-300mm 1:4.0-5.6  Existing lens: Zuiko Digital ED 150mm 1:2.0 |

Your program should correctly **parse the input XML**.

5 score

Your program should correctly **import manufacturers** (new and existing).

7 score

Your program should correctly **import lenses** (new and existing).

12 score

Your program should correctly **add lenses to manufacturer**.

6 score

### \* Generate Random Equipment

Write a **C# application** based on your EF data model for **generating random equipments and cameras**. The application should process **multiple requests** and write **logs** for each operation at the console.

The input comes from an XML file generate-**equipments**.xml in the following format:

|  |
| --- |
| **generate-equipments.xml** |
| <?xml version="1.0" encoding="utf-8" ?>  <generate-random-equipments>  <generate generate-count="2">  <manufacturer>Canon</manufacturer>  </generate>  <generate />  <generate generate-count="4" />  <generate generate-count="3">  <manufacturer>Pentax</manufacturer>  </generate>  <generate generate-count="5">  <manufacturer>Sony</manufacturer>  </generate>  </generate-random-equipments> |

All elements and attributes in the input XML are **non-mandatory**.

* When "**generate-count**" is not specified, it has default value of **10**.
* When "**manufacturer**" is not specified, it has default value of “**Nikon**”.

The output should be printed at the console in the following format:

|  |
| --- |
| Processing request #1 ...  Equipment added: Canon (Camera: PowerShot SX410 IS - Lens: EF 15mm f/2.8 Fisheye)  Equipment added: Canon (Camera: PowerShot SX410 IS - Lens: EF 16-35mm f/2.8L II USM)  Processing request #2 ...  Equipment added: Nikon (Camera: D810 - Lens: AF Nikkor 24-85mm f/2.8-4D IF)  Equipment added: Nikon (Camera: Coolpix L320 - Lens: AF-S Micro-Nikkor 105mm f/2.8G IF-ED VR)  Equipment added: Nikon (Camera: Coolpix P900 - Lens: AF-S Micro-Nikkor 60mm f/2.8G ED)  Equipment added: Nikon (Camera: D5500 - Lens: AF Nikkor 24-85mm f/2.8-4D IF)  Equipment added: Nikon (Camera: Coolpix AW130 - Lens: AF-S DX Nikkor 55-200mm f/4-5.6G VR II)  Equipment added: Nikon (Camera: Coolpix P900 - Lens: AF-S Micro-Nikkor 105mm f/2.8G IF-ED VR)  Equipment added: Nikon (Camera: D5500 - Lens: AF-S Nikkor 24-85mm F3.5-4.5G ED VR)  Equipment added: Nikon (Camera: Coolpix S810c - Lens: AF-S DX Nikkor 55-200mm f/4-5.6G VR II)  Equipment added: Nikon (Camera: Coolpix P530 - Lens: AF-S DX Nikkor 35mm f/1.8G)  Equipment added: Nikon (Camera: D750 - Lens: AF-S DX Nikkor 35mm f/1.8G)  Processing request #3 ...  Equipment added: Nikon (Camera: Coolpix S810c - Lens: AF-S DX Nikkor 35mm f/1.8G)  Equipment added: Nikon (Camera: D750 - Lens: AF-S Micro-Nikkor 60mm f/2.8G ED)  Equipment added: Nikon (Camera: Coolpix AW130 - Lens: AF-S Micro-Nikkor 60mm f/2.8G ED)  Equipment added: Nikon (Camera: D7200 - Lens: AF-S Nikkor 600mm f/4G ED VR)  Processing request #4 ...  Equipment added: Pentax (Camera: K-500 - Lens: DA 21mm F3.2 AL Limited)  Equipment added: Pentax (Camera: K-30 - Lens: DA 21mm F3.2 AL Limited)  Equipment added: Pentax (Camera: Q-S1 - Lens: DA 21mm F3.2 AL Limited)  Processing request #5 ...  Equipment added: Sony (Camera: Alpha QX1 - Lens: 85mm F2.8 SAM)  Equipment added: Sony (Camera: Alpha 7 - Lens: 100mm F2.8 Macro)  Equipment added: Sony (Camera: Cyber-shot DSC-RX100 IV - Lens: 85mm F2.8 SAM)  Equipment added: Sony (Camera: Alpha 7R II - Lens: 70-200mm F2.8 G)  Equipment added: Sony (Camera: Alpha QX1 - Lens: 70-300mm F4.5-5.6 G SSM) |

Your program should correctly **parse the input XML**.

4 score (bonus)

Your program should correctly **generate random equipments**.

12 score (bonus)

Your program should correctly **save in the database** the generated random equipments.

2 score (bonus)

Your program should correctly **print to the console** the generated random matches.

2 score (bonus)

## Part II – EF Code First: Define Data Model, Import and Export Data

You are assigned to define a **code first data model in EF** and write a few data-driven applications in C# for importing data, querying data and exporting data from the database.

Use a new database "**Chat System**". Do not modify the "**Photography**" database.

### EF Code First: Phonebook

Create an **Entity Framework (EF) code first data model** for keeping phonebook holding contacts with phones and emails. It should have several entities:

* **Users** have **username**, and optionally, full name, phone number.
* **Channels** have **name**.
* **ChannelMessages** have **content, datetime** and holds channel and user.
* **UserMessages** have **content**, **datetime** and holds recipient user and sender user.
* Every user can be in many channels.

7 score

**Seed** your database with a few users, channels, and messages, using the **EF migrations** framework. It is OK to drop the database in case of model changes or use any other migration strategy like automatic upgrade to the latest DB schema.

Insert the following **users** in your seed method:

|  |
| --- |
| **Username:** VGeorgiev, **FullName**: Vladimir Georgiev, **PhoneNumber**: 0894545454  **Username:** Nakov, **FullName**: Svetlin Nakov, **PhoneNumber**: 0897878787  **Username:** Ache, **FullName**: Angel Georgiev, **PhoneNumber**: 0897121212  **Username:** Alex, **FullName**: Alexandra Svilarova, **PhoneNumber**: 0894151417  **Username:** Petya, **FullName**: Petya Grozdarska, **PhoneNumber**: 0895464646 |

Insert the following **channels** in your seed method:

|  |
| --- |
| **Name**: Malinki  **Name**: SoftUni  **Name**: Admins  **Name**: Programmers  **Name**: Geeks |

Insert the following **channel messages** in your seed method:

|  |
| --- |
| **Channel**: Malinki, **Content**: Hey dudes, are you ready for tonight?, **DateTime**: Now, **User**: Petya  **Channel**: Malinki, **Content**: Hey Petya, this is the SoftUni chat., **DateTime**: Now, **User**: VGeorgiev  **Channel**: Malinki, **Content**: Hahaha, we are ready!, **DateTime**: Now, **User**: Nakov  **Channel**: Malinki, **Content**: Oh my god. I mean for drinking beers!, **DateTime**: Now, **User**: Petya  **Channel**: Malinki, **Content**: We are sure!, **DateTime**: Now, **User**: VGeorgiev |

5 score

To test your data model, **list all channels along with their messages**.

Example:

|  |
| --- |
| Malinki  -- Messages: --  Content: Hey dudes, are you ready for tonight?, DateTime: 6/23/2015 9:33:19 AM, User: Petya  Content: Hey Petya, this is the SoftUni chat., DateTime: 6/23/2015 9:33:19 AM, User: VGeorgiev  Content: Hahaha, we are ready!, DateTime: 6/23/2015 9:33:19 AM, User: Nakov  Content: Oh my god. I mean for drinking some beer!, DateTime: 6/23/2015 9:33:19 AM, User: Petya  Content: We are sure!, DateTime: 6/23/2015 9:33:19 AM, User: VGeorgiev  SoftUni  -- Messages: --  Admins  -- Messages: --  … |

3 score

### Import User Messages from JSON

Write a **C# application** based on your EF code first data model for **importing into the DB a set of user messages** given in the file messages.json in the following JSON format:

|  |
| --- |
| messages**.json** |
| [  {  "content" : "Hello, I am new here!",  "datetime": "2009-04-12T20:44:55",  "recipient": "Nakov",  "sender": "Ache"  },  {  "content" : "Hi, dude! You are welcome!",  "datetime": "2009-04-12T20:48:35",  "recipient": "Ache",  "sender": "Nakov"  },  {  "content" : "I’m looking for geeks! I want some programming advice.",  "datetime": "2009-04-12T20:51:12",  "recipient": "Nakov",  "sender": "Ache"  },  {  "content" : "Wrong message",  }  {  "content" : "I can help you. I am the geekest geek you have ever seen.",  "datetime": "2009-04-12T20:52:14",  "recipient": "Ache",  "sender": "Nakov"  }  ] |

All properties are mandatory.

You should **parse the JSON** and throw an exception in case of incorrect data, e.g. when a required element is missing or an invalid value is given. The size of the JSON file will be less than **10 MB**. Use a JSON parser by choice.

10 score

You should correctly **import the messages into the DB**.

10 score

Print as **output** a single line for each contact from the input JSON: either "*Message: <content> imported*" or "*Error: <error message>*". Error messages should describe briefly the problem (as free text) and may optionally include exception stack-trace. Sample output:

|  |
| --- |
| Message "Hello, I am new here!" imported  Message "Hi, dude! You are welcome!" imported  Message "I?m looking for geeks! I want some programming advice." imported  Error: Recipient is required  Message "I can help you. I am the geekest geek you have ever seen." imported |

5 score

## Exam Information

You are allowed to use any resources you have, e.g. Internet, software, existing code.

You are not allowed to get help from other people. Skype, ICQ, FB, email, talks, phone calls, etc. are forbidden.

Exam time: **6 hours**.